



⑯ BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

# Offenlegungsschrift

⑰ DE 101 01 067 A 1

⑯ Int. Cl.<sup>7</sup>:  
G 06 F 9/455

⑱ Aktenzeichen: 101 01 067.2  
⑲ Anmeldetag: 11. 1. 2001  
⑳ Offenlegungstag: 16. 8. 2001

DE 101 01 067 A 1

㉑ Unionspriorität:

00-006080 11.01.2000 JP

㉒ Erfinder:

Yamashita, Yasuyoshi, Tokio/Tokyo, JP

㉓ Anmelder:

Advantest Corp., Tokio/Tokyo, JP

㉔ Vertreter:

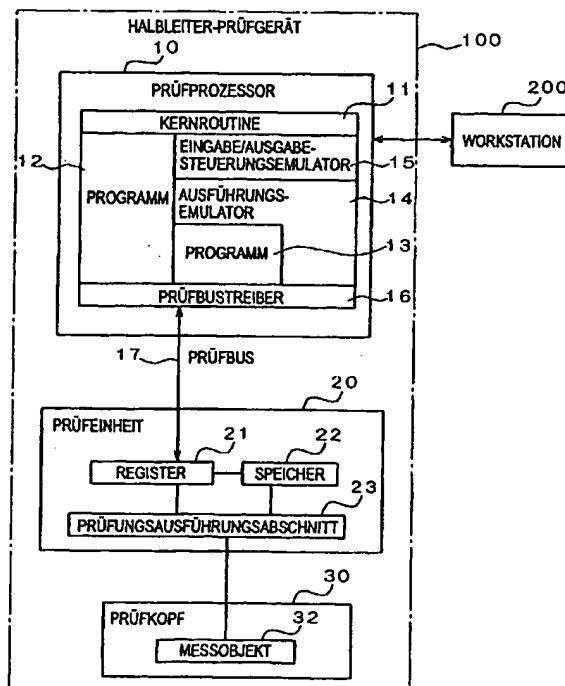
Betten & Resch, 80333 München

### Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

㉕ Programmausführungssystem für ein Halbleiter-Prüfgerät

㉖ Es ist ein Programmausführungssystem für ein Halbleiter-Prüfgerät offenbart. In diesem System enthält ein Vorrichtungs-Prüfprogramm erste Befehle, die in einer universellen Programmiersprache geschrieben sind, und zweite Befehle, die in einer nicht universellen Programmiersprache geschrieben sind, die vom Halbleiter-Prüfgerät abhängt, wobei jeder Befehl bei einem separaten Programm-Ausführungsbeispiel ausgeführt wird.



DE 101 01 067 A 1

## HINTERGRUND DER ERFINDUNG

Diese Erfindung betrifft ein Programmausführungssystem für ein Halbleiter-Prüfgerät, das verschiedene Betriebsprüfungen an einer Halbleitervorrichtung, wie beispielsweise einem Halbleiterspeicher, durchführt.

Herkömmlicherweise sind Halbleiter-Prüfgeräte bekannt, die vorbestimmte Betriebsprüfungen an verschiedenen Halbleitervorrichtungen durchführen. Beispielsweise enthalten die Halbleitervorrichtungen als DUT (Meßobjekt) einen Halbleiterspeicher, eine Logik-IC, eine lineare IC etc., und geeignete Halbleiter-Prüfgeräte werden für jeweilige Halbleitervorrichtungen verwendet. Diese verschiedenen Halbleiter-Prüfgeräte sind entworfen, um vorbestimmte Funktionsprüfungen, Gleichstromprüfungen (DC-Parameterprüfungen) etc. durch Ausführen vorbestimpter durch Anwender programmierte Vorrichtungs-Prüfprogramme durchzuführen. Das Vorrichtungs-Prüfprogramm ist allgemein durch drei Teile gebildet – Prüf- bzw. Test-Steuerbefehle, Datenverarbeitungsbefehle und Algorithmusbefehle. Die Prüf-Steuerbefehle enthalten verschiedene Befehle zum Steuern von Hardware des Halbleiter-Prüfgeräts, wie beispielsweise Befehle zum Einstellen von Prüfbedingungen und Befehle zum Ausführen der Tests bzw. Prüfungen. Die Datenverarbeitungsbefehle haben keinen direkten Bezug zur Hardware des Halbleiter-Prüfgeräts und enthalten Befehle zum Verarbeiten von Daten, die aus den Halbleiterprüfungen resultieren. Die Algorithmusbefehle enthalten Befehle zum Anweisen, wie die Prüfprogramme der gesamten Vorrichtung laufen sollen.

Die oben angegebenen herkömmlichen Vorrichtungs-Prüfprogramme sind in von Herstellern der Halbleiter-Prüfgeräte entwickelten eindeutigen Programmiersprachen geschrieben worden. In einem solchen Vorrichtungs-Prüfprogramm wird ein Objekt als Zwischensprache durch ein Kompilieren erzeugt und ausgeführt. Jedoch deshalb, weil die Befehle des Objekts bei dieser Ausführung Zeile für Zeile interpretiert werden, hat es ein Problem gegeben, daß die Ausführungsgeschwindigkeit verglichen mit den Fällen niedrig ist, in welchen universelle Programmiersprachen, wie beispielsweise die Sprache C, verwendet werden.

Weiterhin hat es für Programmierer von Vorrichtungs-Prüfprogrammen ein Problem gegeben, daß ihre Vertrautheit mit ihrer Programmieraufgabe nicht Zwecken dient, die andere als ein Programmieren ihrer Vorrichtungs-Prüfprogramme sind, so daß sie nicht zu einer Beherrschung einer Programmiererfahrung in universellen Programmiersprachen führt.

Es hat auch ein derartiges Problem gegeben, daß es deshalb, weil die Algorithmusbefehle des Vorrichtungs-Prüfprogramms in einer nicht universellen eindeutigen Programmiersprache geschrieben sind, nicht einfach ist, das Programm in eine universelle Programmiersprache, wie beispielsweise die Sprache C, umzuschreiben, was in einer schlechten Übertragbarkeit auf universelle Programmiersprachen resultiert.

Weiterhin ist deshalb, weil in Vorrichtungs-Prüfprogrammen allgemein Programmiersprachen für begrenzte Zwecke verwendet werden, eine verfügbare Funktionalität verglichen mit den Fällen oft begrenzt, in welchen universelle Programmiersprachen, wie beispielsweise die Sprache C, verwendet werden. Beispielsweise kann ein Ansatz einer strukturierten Programmierung nicht verwendet werden, wenn Strukturen und Vereinigungen nicht vorgesehen sind.

## ZUSAMMENFASSUNG DER ERFINDUNG

Die vorliegende Erfindung wurde angesichts dieser Situationen geschaffen, und es ist eine Aufgabe der vorliegenden

- 5 Erfindung, ein Programmausführungssystem für ein Halbleiter-Prüfgerät zu schaffen, das eine Ausführungsgeschwindigkeit erhöhen kann, das hilft, eine universelle Programmiersprache zu beherrschen, und das eine Übertragbarkeit des Programms verbessern kann.
- 10 Bei dem Programmausführungssystem für ein Halbleiter-Prüfgerät der vorliegenden Erfindung enthält ein Vorrichtungs-Prüfprogramm erste Befehle, die vom Halbleiter-Prüfgerät abhängen und in einer nicht universellen Programmiersprache geschrieben sind, und zweite Befehle, die unabhängig vom Halbleiter-Prüfgerät sind und in einer universellen Programmiersprache geschrieben sind. Das System führt mit dem Halbleiter-Prüfgerät verschiedene Tests bzw. Prüfungen an einer Halbleitervorrichtung durch, indem die oben angegebenen ersten Befehle mit einer ersten Programmausführungseinheit ausgeführt werden und indem die oben angegebenen zweiten Befehle mit einer zweiten Programmausführungseinheit ausgeführt werden. Da die zweiten Befehle, die ein Teil des Vorrichtungs-Prüfprogramms und unabhängig vom Halbleiter-Prüfgerät sind, in einer universellen Programmiersprache geschrieben werden können, ist es einfach, das Vorrichtungs-Prüfprogramm in die universelle Programmiersprache umzuschreiben, so daß eine Übertragbarkeit des Programms verbessert werden kann. Weiterhin kann man zu einer Beherrschung einer Programmiererfahrung in der universellen Programmiersprache gelangen, indem man vertraut mit einem Programmieren etc. des Vorrichtungs-Prüfprogramms wird, und dies läßt zu, daß man ein vielseitiges Wissen erlangt und seine Erfahrung verbessert. Die Ausführungsgeschwindigkeit des gesamten
- 15 Vorrichtungs-Prüfprogramms kann auch erhöht werden, da eine Ausführung von Befehlen, die in einer universellen Programmiersprache, wie beispielsweise der Sprache C, geschrieben sind, allgemein schneller als eine Ausführung von Befehlen ist, die in einer Programmiersprache geschrieben sind, die von einem Halbleiter-Prüfgerät abhängt und eindeutig für dieses ist.
- 20 Weiterhin ist es wünschenswert, daß die oben angegebenen ersten Befehle Befehle zum Steuern des Betriebs des Halbleiter-Prüfgeräts enthalten und daß die zweiten Befehle
- 25 Befehle zum Verarbeiten von Daten enthalten, die aus einer Ausführung der ersten Befehle resultieren, und Befehle zum Definieren von Ausführungsprozeduren des gesamten Vorrichtungs-Prüfprogramms. Da ein Schreiben der Befehle zum Steuern des Betriebs des Halbleiter-Prüfgeräts in einer universellen Programmiersprache in einem unwirtschaftlichen und redundanten Beschreibungsinhalt resultieren würde, kann ein Schreiben der Befehle in einer nicht universellen Programmiersprache verlassen, daß der Inhalt des Vorrichtungs-Prüfprogramms einfach ist und einfach zu verstehen ist. Weiterhin ist es deshalb, weil irgendeine Programmiersprache zum Beschreiben der Teile für eine Beschreibung von Algorithmen, die andere Ausführungsprozeduren anzeigen, und für eine Beschreibung von Inhalten einer Datenverarbeitung verwendet werden kann, möglich,
- 30 35 40

- 45 Befehle zum Verarbeiten von Daten enthalten, die aus einer Ausführung der ersten Befehle resultieren, und Befehle zum Definieren von Ausführungsprozeduren des gesamten Vorrichtungs-Prüfprogramms. Da ein Schreiben der Befehle zum Steuern des Betriebs des Halbleiter-Prüfgeräts in einer universellen Programmiersprache in einem unwirtschaftlichen und redundanten Beschreibungsinhalt resultieren würde, kann ein Schreiben der Befehle in einer nicht universellen Programmiersprache verlassen, daß der Inhalt des Vorrichtungs-Prüfprogramms einfach ist und einfach zu verstehen ist. Weiterhin ist es deshalb, weil irgendeine Programmiersprache zum Beschreiben der Teile für eine Beschreibung von Algorithmen, die andere Ausführungsprozeduren anzeigen, und für eine Beschreibung von Inhalten einer Datenverarbeitung verwendet werden kann, möglich,
- 50 55 60

- 65 Es ist auch wünschenswert, daß das oben angegebene Halbleiter-Prüfgerät eine Prüfungsausführungseinheit zum Erzeugen verschiedener Prüfsignale für die Halbleitervorrichtung und zum Erlangen von Ausgangssignalen, die von der Halbleitervorrichtung in Antwort auf die Prüfsignale

ausgegeben werden, aufweist, und daß das Halbleiter-Prüfgerät eine Eingabe/Ausgabe-Operation der Prüfsignale und der Ausgangssignale durch die Prüfungsausführungseinheit ausführt, wenn es die ersten Befehle durch die erste Programmausführungseinheit ausführt. Ein Durchführen verschiedener Prüfungen an der Halbleitervorrichtung erfordert, verschiedene Signale zwischen dem Halbleiter-Prüfgerät und der Halbleitervorrichtung als Meßobjekt einzugeben/auszugeben, und darüber hinaus ist eine solche Eingabe/Ausgabe-Operation möglich, indem veranlaßt wird, daß die Prüfungsausführungseinheit, die für das Halbleiter-Prüfgerät spezifisch ist, eine spezielle Operation durchführt. Daraus kann durch Verwenden der ersten Befehle, die die eindeutigen Befehle in einer nicht universellen Programmiersprache für eine spezielle Operation enthalten, die auftreten soll, ein effizientes Vorrichtungs-Prüfprogramm programmiert werden.

Es ist auch wünschenswert, daß das oben angegebene Halbleiter-Prüfgerät einen Prüfprozessor hat, der ein erstes Programm, das aus den ersten Befehlen besteht, mit einem vorbestimmten Emulator interpretiert und ausführt, während er ein zweites Programm, das aus den zweiten Befehlen besteht, direkt ausführt. Allgemein ist eine Verarbeitungsgeschwindigkeit bei einer direkten Ausführung des in einer universellen Programmiersprache geschriebenen ersten Programms schneller als diejenige bei einer Ausführung des in einer nicht universellen Programmiersprache geschriebenen ersten Programms, während es mit einem vorbestimmten Emulator interpretiert wird. Somit kann eine Ausführungs geschwindigkeit verglichen mit dem Fall einer Verwendung von herkömmlichen Vorrichtungs-Prüfprogrammen schneller sein, die vollständig in nicht universellen eindeutigen Programmiersprachen geschrieben sind.

Es ist auch wünschenswert, daß die oben angegebene universelle Programmiersprache die Sprache C ist. Die in der Sprache C geschriebenen zweiten Befehle können direkt in diejenigen in einer Assemblersprache übersetzt werden, wenn sie kompiliert sind, so daß eine Ausführungs geschwindigkeit schneller sein kann, weil keine Zwischen sprache verwendet wird, d. h. keine Verarbeitung, wie beispielsweise eine Interpretation der Zwischensprache, erforderlich ist, wenn das Vorrichtungs-Prüfprogramm (Objekt programm) entsprechend den zweiten Befehlen ausgeführt wird.

#### KURZE BESCHREIBUNG DER ZEICHNUNGEN

**Fig. 1** zeigt einen Gesamtaufbau des Halbleiter-Prüfgeräts gemäß einem Ausführungsbeispiel.

**Fig. 2** ist ein Diagramm, das ein Beispiel des Vorrichtungs-Prüfprogramms zeigt.

#### BESCHREIBUNG DES BEVORZUGTEN AUSFÜHRUNGSBEISPIELS

Nun wird ein Halbleiter-Prüfgerät eines Ausführungsbeispiels, auf welches diese Erfindung angewendet wird, unter Bezugnahme auf die Zeichnungen beschrieben.

**Fig. 1** zeigt den Gesamtaufbau des Halbleiter-Prüfgeräts dieses Ausführungsbeispiels. Das in dieser Figur gezeigte Halbleiter-Prüfgerät 100 weist einen Prüfprozessor 10, eine Prüfeinheit bzw. einen Tester 20 und einen Prüfkopf 30 auf. Dieses Halbleiter-Prüfgerät 100 ist konfiguriert, um vorbestimmte Prüfungen bzw. Tests an einem DUT 32, das eine zu prüfende Halbleitervorrichtung ist, unter Verwendung des Prüfkopfs 30 durchführen zu können.

Der Prüfprozessor 10 ist entworfen, um einen Betrieb der Prüfeinheit 20 zu steuern, und weist eine Kernroutine 11,

Programme 12 und 13, einen Ausführungsimulator 14, einen I/O-Steueremulator 15 und einen Prüfbustreiber 16 auf. Eines der Programme, nämlich das Programm 12, ist in der Sprache C geschrieben, welches eine universelle Programmiersprache ist. Das andere Programm, nämlich das Programm 13, ist in einer für diese Halbleiterprüfungen entwickelten nicht universellen eindeutigen Programmiersprache geschrieben. Die zwei Programme 12 und 13 bilden ein Vorrichtungs-Prüfprogramm, das Prozeduren und Details zum Durchführen verschiedener Prüfungen, wie beispielsweise Funktionsprüfungen oder DC- bzw. Gleichstrom-Parameterprüfungen, am DUT 32 definiert.

Die Kernroutine 11 ist ein Echtzeit-Betriebssystem, das Funktionen zum Ausführen des Programms 12, des Ausführungsimulators 14 bzw. des I/O-Steueremulators 15 hat.

Das Vorrichtungs-Prüfprogramm enthält drei funktionelle Teile, nämlich (1) einen Prüfsteuerteil, (2) einen Datenverarbeitungsteil und (3) einen Algorithmusbeschreibungsteil. Unter diesen Teilen besteht der Prüfsteuerteil (1) aus Befehlen, die Befehle zum Steuern von Hardware des Halbleiter-Prüfgeräts 100 enthalten. Der Datenverarbeitungsteil (2) hat keinen direkten Bezug zur Hardware des Halbleiter-Prüfgeräts 100 und besteht aus Befehlen, die Befehle zum Verarbeiten verschiedener Daten enthalten, die als Prüfergebnisse bekommen werden. Der Algorithmusbeschreibungsteil (3) besteht aus Befehlen, die Befehle zum Anzeigen, wie das gesamte Vorrichtungs-Prüfprogramm auszuführen ist, enthalten.

Eines der Programme, nämlich das Programm 12, das in der Sprache C geschrieben ist, entspricht dem Datenverarbeitungsteil und dem Algorithmusbeschreibungsteil unter den drei funktionellen Teilen, die im Vorrichtungs-Prüfprogramm enthalten sind. Die Kernroutine 11 führt dieses Programm 12 direkt aus, um verschiedene Arten von Datenverarbeitungs- und Ausführungsprozeduren zu steuern, die in jedem Befehl enthalten sind.

Das andere Programm, nämlich das Programm 13, das in einer nicht universellen Programmiersprache geschrieben ist, entspricht dem Prüfsteuerteil unter den drei funktionellen Teilen, die im Vorrichtungs-Prüfprogramm enthalten sind.

Der Ausführungsimulator 14 ist entworfen, um das Programm 13 auszuführen, und er interpretiert eine Vielzahl von Zeilen der Befehle, die im Programm 13 enthalten sind, Zeile für Zeile und führt sie aus. Beispielsweise ist das Programm 13 ein Objekt in einer Zwischensprache, die aus der Kompilierung eines Quellprogramms resultiert, und jeder Befehl des Objekts wird durch den Ausführungsimulator 14 interpretiert und ausgeführt.

Der I/O-Steueremulator 15 interpretiert Eingabe/Ausgabe-Befehle und führt sie aus, um eine Eingabe/Ausgabe-Operation von Daten etc. zwischen dem Ausführungsimulator 14 und einem Arbeitsplatzrechner bzw. einer Workstation 200 zu steuern. Die Befehle im Programm 13 enthalten auch Eingabe/Ausgabe-Befehle, die zum Durchführen eines Platten- bzw. Diskettenzugriffs, einer Tastatureingabe oder einer Anzeige für die Workstation 200 erforderlich sind, und die Ausführung dieser Eingabe/Ausgabe-Befehle durch den I/O-Steueremulator 15 gibt Betriebsanweisungen zur Workstation 200. Der Prüfbustreiber 16 ist entworfen, um verschiedene Daten über einen Prüfbus 17 zu senden/zu empfangen, und er steuert ein Senden verschiedener Daten, die für die Funktionsprüfungen oder die DC-Parameterprüfungen erforderlich sind, zur Prüfeinheit 20 oder ein Empfangen von Prüfergebnissen, die von der Prüfeinheit 20 ausgegeben werden.

Die Prüfeinheit 20 ist entworfen, um unter der Steuerung des Prüfprozesses 10 verschiedene Prüfungen, wie bei-

spielsweise Funktionsprüfungen, DC-Parameterprüfungen oder RF-Prüfungen (Radiofrequenz-Prüfungen), für das am Prüfkopf 30 angebrachte DUT 32 durchzuführen. Diese Prüfeinheit 20 weist ein Register 21, einen Speicher 22, und einen Prüfungsausführungsabschnitt 23 auf. Das Register 21 speichert verschiedene Daten, die vom Prüfbustreiber 16 des Prüfprozessors 10 gesendet bzw. von diesem empfangen werden. Im Register 21 gespeicherte Daten werden direkt oder über den Speicher 22 zum Prüfungsausführungsabschnitt 23 gesendet. Vom Prüfungsausführungsabschnitt 23 ausgegebene Daten werden einmal im Register 21 oder im Speicher 22 gespeichert und dann innerhalb des Prüfprozessors 10 über das Register 21 zum Prüfbustreiber 16 gesendet. Der Prüfungsausführungsabschnitt 23, der verschiedene Anordnungen (beispielsweise einen Mustergenerator, einen Zeitgabegenerator, eine DC-Einheit, etc.) enthält und der zum Durchführen von Funktionsprüfungen für das DUT 32 erforderlich ist, erzeugt verschiedene Signale, die zum DUT 32 einzugeben sind, während Daten gemessen werden, die an den Ausgabe-Stiften des DUT 32 erscheinen.

Die oben beschriebene Kernroutine 11 entspricht einer zweiten Programmausführungseinheit, und die Kernroutine 11, der Ausführungsimulator 14 und der I/O-Steuermulator 15 entsprechen einer ersten Programmausführungseinheit, und der Prüfungsausführungsabschnitt 23 entspricht einer Prüfungsausführungseinheit.

Das Halbleiter-Prüfgerät 100 dieses Ausführungsbeispiels hat eine solche Konfiguration, und nun wird sein Betrieb beschrieben. Fig. 2 zeigt spezifische Inhalte der zwei Programme 12 und 13, die das Vorrichtungs-Prüfprogramm bilden. Pfeile zeigen den Programmablauf an und in Klammern gesetzte Zahlen bei den in Fig. 2 gezeigten Pfeilen zeigen die Ablauffolge an. Im folgenden werden die Operationen beschrieben, die dann durchgeführt werden, wenn die jeweiligen Befehle in der Reihenfolge der in Klammern gesetzten Zahlen ausgeführt werden.

(1) Wenn das Vorrichtungs-Prüfprogramm angewiesen wird, ausgeführt zu werden, wie beispielsweise durch Verwenden einer Tastatur, die für die Workstation 200 vorgesehen ist, liest die Kernroutine 11 zuerst das Programm 12 aus, um es sequentiell ab seinem ersten Befehl "main()" auszuführen. Diese "main"-Funktion wird zuerst unter verschiedenen Funktionen der Sprache C, die im in der Sprache C geschriebenen Programm 12 enthalten sind, ausgeführt.

(2) Auf eine Ausführung eines Befehls "execute-ATL("PRO SAMPLE", "initial") durch die Kernroutine 11 hin, ruft das Programm 12 das Programm 13 auf, das durch einen Programmnamen "PRO SAMPLE" spezifiziert ist, um einen Initialisierungsprozeß des Programms anzugeben. In Antwort auf diese Anweisung führt der Ausführungsimulator 14 einen Initialisierungsprozeß des Programms 13 aus, das durch den Programmnamen "PRO SAMPLE" spezifiziert ist.

(3) Nach einer Beendigung des Initialisierungsprozesses führt der Ausführungsimulator 14 einen Befehl "RETURN C" im Programm 13 aus, und eine Ausführungposition kehrt bei einer Position zurück zum Programm 12, bei welcher es zuvor unterbrochen ist.

(4), (5) Auf eine Ausführung eines Befehls "execute-ATL("PRO SAMPLE", "TEST2") durch die Kernroutine 11 hin führt der Ausführungsimulator 14 Befehle aus, die durch "TEST2" des Programms 13 spezifiziert sind. Beispielsweise wird jeder der Befehle "RATE = 10NS" und "MEAS MPAT PAT2" ausgeführt. "RATE" ist ein Befehl zum Einstellen eines Grundzyklus einer Daten-Eingabe/Ausgabe-Zeitgabe. "MEAS MPAT" ist ein Befehl zum Anzeigen eines Beginns einer Messung für die Funktionsprüfung. Da eine Ausführung dieser Befehle Eingabeanal-

nen bestimmter Daten zur Prüfeinheit 20 enthält, wird eine Steuerung zum Prüfbustreiber 16 übergeben, um diese Befehle durch den Prüfbustreiber 16 auszuführen.

(6) Darauffolgend führt der Ausführungsimulator 14 einen Befehl "RETURN C" im Programm 13 aus, und die Ausführungposition kehrt bei einer Position zurück zum Programm 12, bei welcher es zuvor unterbrochen ist.

(7), (8) Auf eine Ausführung eines Befehls "execute-ATL("PRO SAMPLE", "TEST3")" durch die Kernroutine 11 hin, führt der Ausführungsimulator 14 einen Befehl aus, der durch "TEST3" des Programms 13 spezifiziert ist. Beispielsweise führt der Ausführungsimulator 14 einen Befehl "STOP" aus, um einen Beendigungsprozeß der vorbestimmten Funktionsprüfung durchzuführen.

Wie oben ist bei dem Halbleiter-Prüfgerät 100 dieses Ausführungsbeispiels das gesamte Vorrichtungs-Prüfprogramm aus dem Programm 12, das in einer Sprache C geschrieben ist, die eine universelle Programmiersprache ist, und dem Programm 13, das in der nicht universellen Programmiersprache geschrieben ist, die für die Halbleiterprüfungen entwickelt ist, zusammengesetzt. Ein Schreiben der Teile in einer universellen Programmiersprache, die andere als diejenigen zum Anwenden verschiedener Operationen etc. zur Prüfeinheit 20 sind, ermöglicht bzw. erleichtert ein Umschreiben des Vorrichtungs-Prüfprogramms in die universelle Programmiersprache, wodurch die Übertragbarkeit des Programms verbessert wird.

Weiterhin kann man, indem man mit einem Programmieren etc. des Vorrichtungs-Prüfprogramms vertraut wird, die universelle Programmiersprache durch Programmieren des Programms 12, das in der universellen Programmiersprache geschrieben ist, die als Teil des Programms enthalten ist, beherrschen, so daß man ein allgemeines Wissen erlangen und seine Erfahrung verbessern kann. Weiterhin kann deshalb, weil eine Ausführung des Programms 12, das in einer universellen Programmiersprache, wie beispielsweise der Sprache C, geschrieben ist, im allgemeinen schneller als eine Ausführung des Programms 13 ist, das in einer Programmiersprache geschrieben ist, die von Hardware abhängt und eindeutig für diese ist, eine Ausführungsgeschwindigkeit schneller als diejenige von herkömmlichen Vorrichtungs-Prüfprogrammen sein, deren Gesamtheit in nicht universellen eindeutigen Programmiersprachen geschrieben worden ist.

Zusätzlich ermöglicht ein Schreiben des Programms 12 in der Sprache C eine Verwendung von Strukturen und Vereinigungen, die für die Sprache C vorgesehen sind, um einen Ansatz einer strukturierten Programmierung zu verwenden. Diese Erfindung soll nicht auf das oben beschriebene Ausführungsbeispiel beschränkt sein, sondern verschiedene Modifikationen sind innerhalb des Schutzmangels dieser Erfindung möglich. Beispielsweise können, obwohl beim obigen Ausführungsbeispiel das Programm 12 in der Sprache C geschrieben ist, irgendwelche universellen Programmiersprachen verwendet werden, die andere als die Sprache C sind. Beispielsweise kann JAVA (eingetragene Marke) zum Schreiben des Programms 12 verwendet werden.

#### Patentansprüche

1. Programmausführungssystem für ein Halbleiter-Prüfgerät, welches System verschiedene Prüfungen für eine Halbleitervorrichtung unter Verwendung des Halbleiter-Prüfgeräts durch Ausführen eines Vorrichtungs-Prüfprogramms durchführt, wobei das System folgendes aufweist:  
eine erste Programmausführungseinheit zum Ausführen erster Befehle, die vom Halbleiter-Prüfgerät abhän-

gen, wobei die ersten Befehle im Vorrichtungs-Prüfprogramm enthalten sind und in einer nicht universellen Programmiersprache geschrieben sind, und eine zweite Programmausführungseinheit zum Ausführen zweiter Befehle, die unabhängig vom Halbleiter-Prüfgerät sind, wobei die zweiten Befehle im Vorrichtungs-Prüfprogramm enthalten sind und in einer universellen Programmiersprache geschrieben sind. 5

2. Programmausführungssystem für ein Halbleiter-Prüfgerät nach Anspruch 1, wobei:  
die ersten Befehle Befehle zum Steuern eines Betriebs des Halbleiter-Prüfgeräts enthalten, und  
die zweiten Befehle zum Verarbeiten von Daten enthalten, die aus einer Ausführung der ersten Befehle resultieren, und Befehle zum Definieren von Ausführungsprozeduren des gesamten Vorrichtungs-Prüfprogramms. 10 15

3. Programmausführungssystem für ein Halbleiter-Prüfgerät nach Anspruch 1 oder 2, wobei:  
das Halbleiter-Prüfgerät eine Prüfungsausführungseinheit zum Erzeugen verschiedener Prüfsignale für die Halbleitervorrichtung und zum Erlangen von Ausgangssignalen, die von der Halbleitervorrichtung in Antwort auf die Prüfsignale ausgegeben werden, aufweist, und 20 25

Eingabe/Ausgabe-Operationen der Prüfsignale und der Ausgangssignale durch die Prüfungsausführungseinheit ausgeführt werden, wenn die erste Programmausführungseinheit die ersten Befehle ausführt.

4. Programmausführungssystem für ein Halbleiter-Prüfgerät nach einem der Ansprüche 1 bis 3, wobei das Halbleiter-Prüfgerät einen Prüfprozessor hat, der ein erstes Programm, das durch die ersten Befehle gebildet ist, mit einem vorbestimmten Emulator interpretiert und ausführt, während er ein zweites Programm, das durch die zweiten Befehle gebildet ist, direkt ausführt. 30 35

5. Programmausführungssystem für ein Halbleiter-Prüfgerät nach einem der Ansprüche 1 bis 4, wobei die universelle Programmiersprache die Sprache C ist.

6. Halbleiter-Prüfgerät, das folgendes aufweist: ein 40 Programmausführungssystem nach einem der Ansprüche 1 bis 5.

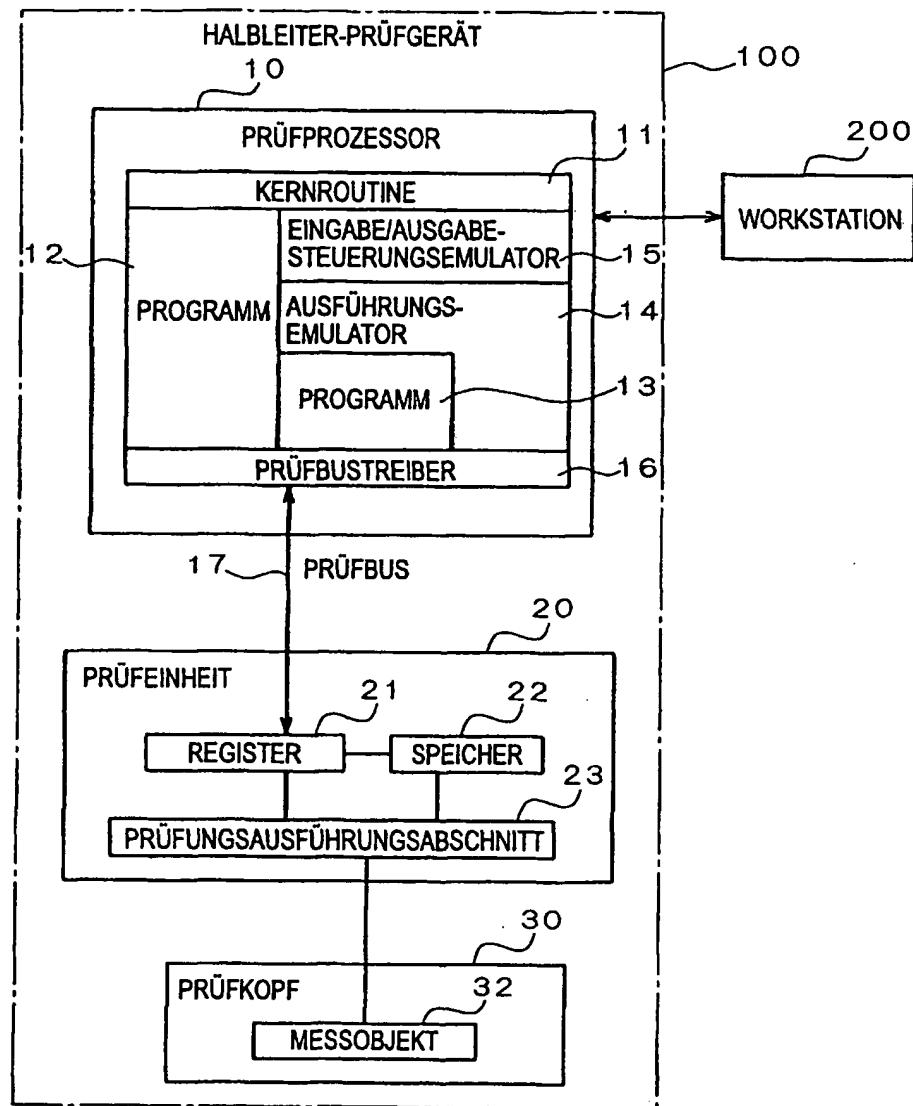
---

Hierzu 2 Seite(n) Zeichnungen

- Leerseite -

**THIS PAGE BLANK (USPTO)**

FIG. 1



F/G. 2

